

AMENDMENTS TO CLAIMS

1 1. (original) A method for translating objects between applications that use different
2 formats, the method comprising:
3 generating a source object in a source application;
4 translating the source object to a target object in a target application, wherein the
5 target application has a format that is not supported by the source application;
6 performing a first modification to the target object, wherein said first modification is
7 not supported by said source application;
8 performing a second modification to said source object in said source application; and
9 revising said target object in said target application to reflect said second modification
10 to said source object without removing said first modification to said target
11 object.

1 2. (original) The method of Claim 1, wherein the step of performing the first
2 modification to the target object includes the step of performing a type of modification
3 that cannot be performed using said source application.

1 3. (original) The method of Claim 1, wherein:
2 the source application is a Computer Aided Design (CAD) application;
3 the target application is a rendering application; and wherein
4 the step of generating the source object in the source application includes the step of
5 generating a CAD object in said CAD application;
6 the step of translating the source object to the target object includes the step of
7 translating the CAD object into a rendering object;

the step of performing the first modification to the target object includes the step of performing a modification to the rendering object;
the step of performing a second modification to said source object includes the step of performing a modification to the CAD object; and
the step of revising said target object includes the step of revising the rendering object to reflect the second modification that was made to the CAD object without undoing the first modification to the rendering object.

4. (original) The method of Claim 1, wherein:

the source object is associated with a source geometry and one or more source properties; and
the step of translating the source object to the target object includes the steps of translating the source geometry to a target geometry; and
translating the one or more source properties to one or more target properties.

5. (original) The method of Claim 1, wherein the step of translating the source object to the target object includes the step of:
building a mapping based on a translation between the source object and the target object.

6. (original) The method of Claim 5, wherein the step of building the mapping includes the step of:
constructing a hierarchical tree structure, wherein the hierarchical tree structure is based on one or more properties associated with the source object.

1 7. (original) The method of Claim 6, wherein
2 the source object is associated with a source geometry and one or more source
3 properties; and
4 the step of constructing the hierarchical tree structure includes the steps of:
5 generating a set of tree objects, wherein the set of tree objects include one or
6 more filter objects that are based on said source properties;
7 translating the source geometry to a target geometry; and
8 inserting said target geometry into said hierarchical tree structure based said
9 one or more filter objects.

1 8. (original) The method of Claim 7, wherein the step of generating the set of tree
2 objects includes the steps of:
3 translating the one or more source properties to one or more target properties;
4 generating one or more modifier stacks, wherein the one or more modifier stacks are
5 based on the one or more target properties; and
6 inserting the one or more modifier stacks into the hierarchical tree structure.

1 9. (original) A method for translating objects between applications that use different
2 formats, the method comprising:
3 generating a first object in a first application;
4 translating the first object to a second object in a second application, wherein the
5 second object has a format that is not supported by the first application;
6 performing a first modification to the second object in the second application;
7 performing a second modification to said first object in said first application; and

performing a third modification to the second object based on data generated in response to said second modification to said first object, wherein said third modification causes said second object to reflect the second modification that was made to the first object without undoing the first modification to the second object.

10. (original) The method of Claim 9, wherein the step of performing the first modification to the second object includes the step of performing a type of modification that cannot be performed using said first application.

11. (original) The method of Claim 9, wherein:
the first application is a Computer Aided Design (CAD) application;
the second application is a rendering application; and wherein
the step of generating the first object in the first application includes the step of generating a CAD object in said CAD application;
the step of translating the first object to the second object includes the step of translating the CAD object into a rendering object;
the step of performing the first modification to the second object includes the step of performing a modification to the rendering object;
the step of performing a second modification to said first object includes the step of performing a modification to the CAD object; and
the step of performing the third modification to the second object includes the step of performing a third modification to the rendering object to reflect the second

14 modification that was made to the CAD object without undoing the first
15 modification to the rendering object.

1 12. (original) A computer-readable medium carrying one or more sequences of
2 instructions for translating objects between applications that use different formats,
3 wherein execution of the one or more sequences of instructions by one or more
4 processors causes the one or more processors to perform the steps of:
5 generating a source object in a source application;
6 translating the source object to a target object in a target application, wherein the
7 target application has a format that is not supported by the source application;
8 performing a first modification to the target object, wherein said first modification is
9 not supported by said source application;
10 performing a second modification to said source object in said source application; and
11 revising said target object in said target application to reflect said second modification
12 to said source object without removing said first modification to said target
13 object.

1 13. (original) A system for translating objects between applications that use different
2 formats, the system comprising:
3 a memory;
4 one or more processors coupled to the memory; and
5 a set of computer instructions contained in the memory, the set of computer
6 instruction including computer instructions which when executed by the one or
7 more processors, cause the one or more processors to perform the steps of:

generating a source object in a source application;
translating the source object to a target object in a target application, wherein
the target application has a format that is not supported by the source
application;
performing a first modification to the target object, wherein said first
modification is not supported by said source application;
performing a second modification to said source object in said source
application; and
revising said target object in said target application to reflect said second
modification to said source object without removing said first
modification to said target object.

14. (currently amended) A method for translating objects between applications that use different formats, the method comprising:
generating a hierarchical structure for organizing one or more properties of a source object being translated to a target object, wherein each level of the hierarchical structure is associated with a property of an object and wherein the source object is associated with a source application and the target object is associated with a target application;
using one or more filter objects to determine a location, within the hierarchical structure, to map the one or more properties of the source object, wherein each of the one or more filter objects is associated with a respective level of the hierarchical structure and associated with one or more collection objects of a set of collection objects, comprising

13 determining, for a property of the one or more properties of the source object, a
14 property value from a respective filter object that is associated with the
15 property;
16 comparing the property value with a respective collection value associated with
17 each of one or more respective collection objects of the set of collection
18 objects that are associated with the respective filter object; and
19 determining a level within the hierarchical structure to map the one or more
20 properties of the source object, based on the comparing the property
21 value with a respective collection value; and
22 storing the hierarchical structure in a target file, wherein the target file is used by the
23 second application to construct the target object.

1 15. (canceled)

1 16. (previously presented) The method of claim 14, further comprising:
2 upon a modification of a property of the target object in the target application,
3 generating a modifier stack for storing the modification, wherein the property of
4 the target object is associated with a respective property of the source object;
5 linking the modifier stack with a collection object of a set of collection objects, wherein
6 each collection object of the set of collection objects is associated with a
7 respective level of the hierarchical structure; and
8 applying the modification of the modifier stack to the target file via the linked collection
9 object to construct the target object.

1 17. (previously presented) The method of claim 16, wherein each of the one or more filter
2 objects is associated with a respective level of the hierarchical structure, comprising:

3 upon a modification of a property of the source object in the source application, using a
4 filter object of the one or more filter objects to determine a level within the
5 hierarchical structure to store the modification;
6 applying the modification of the property of the source object to the target file that
7 includes the stored hierarchical structure, at the determined level within the
8 hierarchical structure; and
9 applying the modification of the modifier stack to the target file to construct the target
10 object.